IPng Working Groups
INTERNET-DRAFT

A. Conta (Transwitch)
B. Carpenter (IBM)
July 2001

A proposal for the IPv6 Flow Label

Specification

draft-conta-ipv6-flow-label-02.txt

Status of this Memo

Abstract

At the time when the IPv6 specifications were written, the IPv6 flow label was still experimental, and subject to change, as the requirements for flow support in the Internet were evolving.

The last several years of work in IETF on Internet Protocols Quality of Service (Intserv, and Diffserv) and Multi-Protocol Label Switching (MPLS) provide a more solid and ample architectural perspective, and framework for the standardization of the IPv6 flow label. The new charter of the IPv6 Working Group invites contributions to this standardization.

This memo provides an analysis of the IPv6 definition of the flow label, the rules governing its use, and their implications. It

subsequently makes a proposal for additions/modifications to these rules, which improve the usability of the IPv6 flow label, in particular with Diffserv, and its acceptance as a standard mechanism.

Table of Contents

1. Introduction

As stated by [IPv6], at the time when the IPv6 specifications were written, the IPv6 flow label was still experimental, and subject to change, as the requirements for flow support in the Internet were evolving.

The last several years of work in IETF on Internet Protocols Quality of Service (Intserv, and Diffserv) and Multi-Protocol Label Switching (MPLS) provide a more solid and ample architectural perspective, and framework for the standardization of the IPv6 flow label. The new charter of the IPv6 Working Group invites contributions to this standardization.

Note: The IETF work on Intserv, Diffserv, MPLS is documented in several specifications, among which the architecture documents [Intserv], [Diffserv], and respectively [MPLS-Arch]. Intserv and Diffserv present two alternative solutions to resolving QoS problems in the Internet, while MPLS is a technology based on labeling traffic flows.

The IPv6 flow label is a function that, as it was designed, can be used towards a more efficient processing of packets in next hop lookup, quality of service, or packet filtering engines in IPv6 forwarding devices. These devices would normally be IPv6 routers or switches. However, the current IPv6 flow label definition and specification can be further clarified or even improved, in particular in regards to Differentiated Services Quality of Service (Diffserv).

Diffserv seems to have more potential, and could be used more extensively than originally thought. For instance, for IP QoS in access networks, Diffserv could be used on individual flows of traffic between users and the access networks. The nature of the contractual agreements between the users and the access network providers seem to create an environment in which Diffserv with Multi-Field (M-F) classifiers could be easier to use, more efficient, and more practical as an alternative to Intserv and RSVP.

However, the Diffserv M-F classifiers, the 5 or 6 element tuple, containing host-to-host protocol id, and source and destination ports, is a bit of a problem when packets have extension headers (IPv4, or IPv6). In IPv6, that is even more of an efficiency problem (need for sequential inspection), since extension headers have a much wider and frequent use.

The IPv6 flow label, and the use of IPv6 flow label classifiers would be a big help in alleviating this problem. An IPv6 flow label

classifier is basically a 3 element tuple - source and destination IPv6 addresses, and the IPv6 flow label [Diffserv-Flow-Label]. It is an alternative to the 5 element tuple (addresses, ports, and protocol). It will help the IPv6 flow label to achieve, as it is supposed, a more efficient processing of packets in quality of service engines in IPv6 forwarding devices.

This specification provides an analysis of the definition of the IPv6 flow label [IPv6], the rules governing its use, and attempts to make clarifications to their implications. It subsequently suggests some additions, or modifications to these rules, which in the view of the authors, improve the usability of the IPv6 flow label, in particular with Diffserv, and its acceptance as a standard mechanism.

The keywords MUST, MUST NOT, MAY, OPTIONAL, REQUIRED, RECOMMENDED, SHALL, SHALL NOT, SHOULD, SHOULD NOT are to be interpreted as defined in [KEYWORDS].

2. IPv6 Flows

A flow is a sequence of packets sent from a particular source, and a particular application running on the source host, using a particular host-to-host protocol for the transmission of data over the Internet, to a particular (unicast or multicast) destination, and particular application running on the destination host, or hosts, with a certain set of traffic, and quality of service requirements.

3. Other Definitions of Flows

As IPv6 relies on Quality of Service Mechanisms defined by the Integrated Services Architecture or the Differentiated Services Quality of Service Architecture, it is worth considering those architectures flow definitions. The MPLS architecture also defines a technique of labeling flows worth considering.

3.1 Integrated Services Flows

The Integrated Services architecture [Intserv] defines a flow as an abstraction which is a distinguishable stream of related datagrams that results from a single user activity and requires the same QoS. For example, a flow might consist of one transport connection or one video stream between a given host pair. It is the finest granularity of packet stream distinguishable by the Integrated Services.

Furthermore, the Integrated Services architecture [Intserv] defines a

classifier:

For the purpose of traffic control (and accounting), each incoming packet must be mapped into some class; all packets in the same class get the same treatment from the packet scheduler. This mapping is performed by the classifier. Choice of a class may be based upon the contents of the existing packet header(s) and/or some additional classification number added to each packet.

A class might correspond to a broad category of flows, e.g., all video flows or all flows attributable to a particular organization. On the other hand, a class might hold only a single flow. A class is an abstraction that may be local to a particular router; the same packet may be classified differently by different routers along the path. For example, backbone routers may choose to map many flows into a few aggregated classes, while routers nearer the periphery, where there is much less aggregation, may use a separate class for each flow.

3.2 Differentiated Services Flows

The Differentiated Services architecture [Diffserv] defines a flow or microflow as a single instance of an application-to-application flow of packets, which is identified by the source address, source port, destination address, destination port and protocol id (fields in the IP and host-to-host protocol headers).

Furthermore, this architecture defines a classifier as:

a mechanism that selects packets in a traffic stream based on the content of some portions of the packet header. Two types of classifiers are defined. The BA (Behavior Aggregate) Classifier classifies packets based on the DS codepoint only. The MF (Multi-Field) classifier [Diffserv-Model] selects packets based on the value of a combination of one or more header fields, such as source address, destination address, DS field, protocol ID, source port and destination port numbers, and other information such as incoming interface.

Classifiers are used to "steer" packets matching some specified rule to an element of a traffic conditioner for further processing. Classifiers must be configured by some management procedure in accordance with the appropriate TCA.

Note: For the purpose of this document, only a portion of the definition of the classifier from the architecture [Diffserv] is mentioned.

3.3 MPLS Flows

As it travels from its source to its final destination, an IP packet is being forwarded from one router to the next, each router making an independent forwarding decision (next hop) based on the packet's IP header, and routing information processed and stored. Choosing the next hop can be thought of as the composition of two functions. The first function partitions the entire set of possible packets into a set of "Forwarding Equivalence Classes (FECs)" [MPLS-Arch]. The second maps each FEC to a next hop. Insofar as the forwarding decision is concerned, different packets, which get mapped into the same FEC, are indistinguishable. All packets, which belong to a particular FEC, and which travel from a particular node, will follow the same path (or if certain kinds of multi-path routing are in use, they will all follow one of a set of paths associated with the FEC). In MPLS, the assignment of a particular packet to a particular FEC results in a label being associated to that FEC. When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are "labeled" before they are forwarded. Once a packet is labeled, at subsequent hops, the forwarding is done based on the MPLS label rather than the information in the IP header. The label is used as an index into a table which specifies the next hop, and a new label. The old label is replaced with the new label, and the packet is forwarded to its next hop.

4. IPv6 Flow Label

The IPv6 Flow Label is defined [IPv6] as a 20 bit field in the IPv6 header which may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. According to [IPv6], the nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option.

The characteristics of IPv6 flows and flow labels, or the rules that govern the flow label functions are further defined in [IPv6]. For the purpose of this document the text from one paragraph in [IPv6] was rearranged as an item list, as follows:

(a) A flow is uniquely identified by the combination of a source address and a non-zero flow label.

(b) Packets that do not belong to a flow carry a flow label of zero.

(c) A flow label is assigned to a flow by the flow's source node.

(d) New flow labels must be chosen (pseudo-)randomly and uniformly from the range 1 to FFFFF hex. The purpose of the random allocation is to make any set of bits within the Flow Label field suitable for use as a hash key by routers, for looking up the state associated with the flow.

(e) All packets belonging to the same flow must be sent with the same source address, destination address, and flow label.

(f) If packets of a flow include a Hop-by-Hop Options header, then they all must be originated with the same Hop-by-Hop Options header contents (excluding the Next Header field of the Hop-by-Hop Options header).

(g) If packets of a flow include a Routing header, then they all must be originated with the same contents in all extension headers up to and including the Routing header (excluding the Next Header field in the Routing header).

(h) The routers or destinations are permitted, but not required, to verify that these conditions are satisfied. If a violation is detected, it should be reported to the source by an ICMP Parameter Problem message, Code 0, pointing to the high-order octet of the Flow Label field (i.e., offset 1 within the IPv6 packet).

(i) The maximum lifetime of any flow-handling state established along a flow's path must be specified as part of the description of the state-establishment mechanism, e.g., the resource reservation protocol or the flow-setup hop-by-hop option.

(j) A source must not reuse a flow label for a new flow within the maximum lifetime of any flow-handling state that might have been established for the prior use of that flow label. When a node stops and restarts (e.g., as a result of a "crash"), it must be careful not to use a flow label that it might have used for an earlier flow whose lifetime may not have expired yet.

5. IPv6 Flow and Flow Label Discussion

This section is going to discuss several aspects of the flow label, which are the target of clarifications or improvement.

5.1 Flow Label processing by Integrated Services Routers

The Integrated Services traffic classification based on flow label in conjunction with the use of the Resource Reservation Protocols (RSVP) for propagating the flow label value seem to be in synchronism. This topic does not require further discussion.

The capability to specify a filter based on source, and destination addresses, and flow label presents the advantage of having all the filtering elements in one header, as opposed to multiple headers.

5.2 Flow Label processing by Differentiated Services Routers

At the time of the writing of this document, the Differentiated Services architecture definition of classifiers [Diffserv] does not seem to include, nor to exclude explicitly the classification of IPv6 packets based on flow labels. The definition in [Diffserv-Model] is general enough to invite the use of the flow label.

In order to support the Flow Label, a Differentiated Services IPv6 classifier definition should be added. This classifier would be a multi-field classifier, which would include as classification fields at least the flow label, and the source address, as the IPv6 specification [IPv6] suggests. To allow and use a wild card source address is perhaps debatable. The MF classifier could be extended with the destination address, so it would be a 3 element tuple: source and destination addresses, and flow label. Range of addresses, or range of flow labels may be specified.

The definition of a MF classifier based on source, and destination addresses, and flow label presents the advantage of having all the classification elements in one packet header, as opposed to scattered in one packet's multiple headers, that is, the IPv6 main header, and transport (or host-to-host) header.

According to the Differentiated Services architecture [Diffserv] the classification fields have values according to the Service Level Agreemnts (SALs), and Traffic Conditioning Agreements (TCAs), (Service Level Specifications -- SLSs, and Traffic Conditioning Specifications -- TCSs) which are contractual agreements between network clients and network service providers. The flow label based Diffserv MF classifier would follow the same model, and would rely on

the flow label which is a field with a value or range of values on which clients and service providers would have to agree on. That value, or value ranges of the flow labels would be reflected in SLAs, TCAs, SLSs, and TCSs.

As the Diffserv classifier fields are known a priori, before traffic is being generated by a source of packets, the same should apply to the flow label classifier and the flow label value. This is contradicted by a random generation of the flow label value. In order to resolve this contradiction, rule marked (d) in Section 4, extracted from [IPv6], Appendix A, which states that the flow label should be pseudo-random, must be relaxed or removed (a subsequent section is a summary of proposals).

5.3 Flow Label based Filtering

A similar problem as the Multi-Field classifier contradiction described in the section above occurs with any type of filtering that a forwarding engine may have to perform, in which the filtering rules are configured by a network manager, or are loaded in the forwarding engine by methods other than a resource reservation protocol, or hop by hop signaling. Note that the filtering may have just internal purposes to a forwarding engine, or to a router (which is assumed may have several forwarding engines), or to a segment of the network, or to a network. In all of the cases enumerated above, the expectation, or assumption is that the IPv6 header carries in its fields a set of predictable, or well determined values. This is not the case, if the flow label has a randomly chosen value.

This problem of not being able to configure or load filtering rules, which are based or are including the flow label, can be resolved simply by relaxing or removing the rule marked (d) in Section 4, extracted from [IPv6], Appendix A, which is that the flow label must be a random number.

5.4 End-to-end/Hop-by-hop use of the IPv6 Flow Label

The definition in [IPv6] gives a definite hop-by-hop characteristic to the flow label. The flow label is supposed to help the routing system in processing packets whether during packet forwarding, or whether during QoS processing. However, controversial discussion took place around the end-to-end use and character of the flow label.

For instance it was stated that the label should be used as a mechanism for identifying a flow by the destination end-node. Such statements seem to be warranted by the use of the IPv6 pair of source

and destination addresses as component fields in host-to-host connection (virtual circuit oriented communication) or communication (connectionless oriented) identifiers, and thus the flow label would just be an addition or a replacement to such identifiers. However, if the routers' packet processing is more performance critical then end-nodes' processing, as the author of this document believes, it would seem to make more sense to use the flow label for that purpose, that is to use the flow label hop-by-hop significance.

Using a flow label end-to-end or hop-by-hop seem to be fine in the context of the current definition of the flow label, as long as the non-mutable character of the flow label is maintained. The issue of mutable or non-mutable is going to be discussed in a separate section.

The discussion around the end-to-end, or hop-by-hop use of a flow label becomes irrelevant if a certain negotiation mechanism amongst routers and end-nodes takes place. There are examples of technologies in which such negotiations around flow labels and flows labeling take place. For instance the Label Distribution Protocol of MPLS [MPLS-LDP] is used to exchange labels among neighboring MPLS Routers, including the source and the destination of the labeled packets. Furthermore, the Resource Reservation Protocol (RSVP) [RSVP] has been extended [RSVP-TE] to exchange labels between neighboring label switch (MPLS) routers. But such a mechanism, at the time of writing this specification, does not exist for IPv6 flow labels, or as part of the IPv6 set of specifications. However, such a mechanism could be specified in the future, therefore the specification or the definition of the IPv6 flow label should not restrict the use of the flow label in one way or another relative to its end-to-end or hop-by-hop characteristic.

In conclusion, the flow label could have a bivalent character in the type of its usage, or in its significance:

(i) end-to-end, and

(ii) hop-by-hop.

The end-to-end significance should not preclude its hop-by-hop significance, and vice-versa. If a node which sends packets, associates a certain end-to-end significance to the flow label of those packets, that significance can be meaningful also hop-by-hop to each downstream router, all the way to the final destination. Furthermore, the flow label could be changed in the packet headers by the en-route routers, and restored or not to its original value by the last hop router, as long as the end-node is aware of what the value of the flow label should be. Certainly such a behavior would

need negotiation and state storing in the en-route routers, in particular the last hop one.

5.5 Mutable/Non-mutable IPv6 Flow Label

Another topic of controversial discussion is whether the flow label should be mutable or non-mutable, that is it should be read-only for routers or not.

Statements that advocate a non-mutable characteristic are certainly based on the advantage of the simplicity implied by such a characteristic.

Opposite statements, that the flow label should be mutable, are based on the flexibility that this provides, in particular if the label has a hop-by-hop significance. However, using mutable flow labels would not work without a certain agreement, or negotiation between neighboring nodes (routers), or certain configuration of those routers. This would require the use of a negotiation mechanism between neighboring routers, or a certain setup through router management or configuration, to make sure that the values or the changes made to the flow label are known to all routers on the portion of the path of the packet, in which the flow label changes. Some of these mechanisms, such as MPLS Label Distribution Protocol [MPLS-LDP], or RSVP extensions for Traffic Engineering [RSVP-TE], were briefly mentioned in the previous section. Such a mechanism could be specified for IPv6 flow labels.

As the hop-by-hop significance of the flow label can be enhanced by a mutable characteristic, the specification or definition of the flow label should not preclude this.

A mutable flow label though requires the relaxation or elimination of the rules marked (a), (c), (d), and (j) in Section 4. These rules were extracted from [IPv6], Appendix A.

5.6 Using Random Numbers in setting the IPv6 Flow Label

The rule marked (d) in Section 4, extracted from [IPv6], Appendix A, specifies the requirement of pseudo-randomness in setting the value of a flow label. The reason given is the use of a hashing function, and hashing table for flow lookup by routers. Randomness certainly helps if the flow label is the only criterion used in the flow lookup.

The use of a hashing mechanism is one possible choice for the flow

lookup in routers, or hosts.

Another possible choice is to use the label as an index in an array, which is a direct and faster lookup, or retrieval of the flow state, and so a contiguous set of values, starting from 1, would be more helpful, in particular if the flow label is not the only criterion used.

However, the authors of this document believe that the specification of the flow label should not mandate any implementation choices, whether they are random values, with hashing functions, or just contiguous values, with array indexing.

Furthermore, a random value in the header is introducing the unpredictability of the field. Although this may be an argument of philosophical nature, predictability is a necessary condition for deterministic behavior. Deterministic behavior is a MUST in a network. Network operators may require that packets of a flow have always the same IPv6 content. Random values in the IPv6 flow label certainly break such a requirement.

To resolve these issues would certainly require the relaxation or elimination of rule marked (d), in Section 4, extracted from Appendix A of [IPv6].

5.7 IPv6 Multi-Field Classifiers Efficiency

This section will address multi-field classification engines efficiency issues.

5.7.1 Classification Rules Memory Requirements

When the flow label value is completely independent from host-to-host protocol id and source and destination port information, the classification rules that contain MF flow label classifiers are at least partially independent from the classification rules that contain regular MF classifiers. If somewhat the flow label could capture the port and host-to-host protocol information, then the flow label classifier values could be in their entirety inferred from a regular M-F classifier values. This could help in storing classification rules in encoding, and perhaps aggregating information in ways in which memory consumption could be minimized. However, the issue and the gain could be categorized as minor.

5.7.2 Pipe-Lined or Parallel Processing Classification.

As it was stated above, an IPv4 QoS multi-field classification engine, performs a lookup of 5 or 6 fields of the IP and Host-to-host protocol headers, in the classification rules table. As most of the time, these headers are back to back (contiguous), the position of the fields is well-known, and therefore the processing can be pipelined or parallelized efficiently. Certainly, the existence of one or more IPv4 security headers, disturbs the contiguity of the headers, but as an encrypted packet would have the host-to-host header encrypted, it is likely that its fields would not be part of a classification rule for that packet's flow.

In IPv6, in case of a Multi-Field Classifier, the IPv6 extension headers that are potentially located between the IPv6 header and the host-to-host protocol header, need to be processed sequentially, before having access to the host-to-host protocol id, and the host-to-host source and destination ports. This adds a certain degree of difficulty in designing a pipe-lining or parallel processing mechanism. The use of the flow label as a replacement of the host-by-host fields (source and destination ports and protocol id) in the classification rules certainly alleviates this issue. Furthermore, the use of the flow label, relaxes the issue mentioned previously with security headers.

6. Summary of Proposals for the IPv6 Flow Label

In summary, the following are the actions being proposed:

1. For the Differentiated Services M-F Classification rules to include the IPv6 flow label classifer:

(i) Write a document that defines a flow label based classifier. This is going to be a separate document, a Differentiated Services specification.

(ii)Make a slight change to the flow label definition, by introducing the Diffserv flow label format.

(iii)Rules in Appendix A of [IPv6], do not apply to Diffserv IPv6 flow labels.

2. For the Diffserv IPv6 flow labels:

(i) Redefine characteristics or rules (a), (b), (c), (i), (j) for Diffserv IPv6 Flow Labels.

(ii) Remove characteristics (e), (f), (g) for Diffserv IPv6 flow labels. They prevent certain ways of aggregating flows into one flow.

The following section, contains the text that specifies the newly suggested IPv6 flow label definition and rules. They would apply to Diffserv flows, and to the use of flow label based non-QoS filtering. They could also apply to Intserv flows, since there is no technical reason that would prevent that.

7. IPv6 Flow Label Definition and Characteristics

The IPv6 Flow Label is a 20 bit field in the IPv6 header which may be used to label packets of the same packet flow, or aggregation of flows. This labeling can be used by IPv6 Quality of Service engines in routers, for packet classification, policing, and scheduling. It can also be used by IPv6 filtering engines in routers, that use filtering for various purposes. Documenting such filtering purposes is beyond the scope of this document.

The flow label values can be communicated to routers through a resource reservation protocol, by a flow label distribution protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option. They can also be configured in routers, manually, or by ways of some automated procedures, or simply uploaded through management or policy control procedures.

The characteristics of IPv6 flows and flow labels are further defined as:

(a) A flow is uniquely identified by the combination of source address, destination address and a non-zero flow label. Diffserv flows MAY be aggregated by specifying a range of addresses and/or a range of flow labels (see further in (e)).

(b) A flow label of zero means that the flow label has no significance, the field is unused, and therefore has no effect on, or for the packet processing by forwarding, QOS, or filtering engines.

(c) A flow label is assigned to a flow by the flow's source node. It can be changed en-route, with the condition that its original significance be maintained, or restored, when necessary. For instance if the source of the flow intended that the flow label

has a certain significance to the destination end-node, than the nodes en-route, that process and eventually change the value of the flow label, should make sure, in conjunction with the destination end-node, that even when the value or significance has changed en-route, the original information and significance is restored when or before the packet arrives to its destination.

If the action to be performed on a particular flow label is not known, a router MUST not change the value of that flow label.

(d) The flow label must have a value between 1 and FFFFF in hex. It identifies a flow. It is a preset value. No particular method is preferred for choosing the value. However, the value MUST satisfy the following requirements:

(i) It can be communicated to all routers on the path of the flow to the final destination, as well as the destination node, by ways of a resource reservation protocol, a flow label distribution protocol, a signaling mechanism, or by any other means. The first method is typical for the Integrated Services model.

(ii) It can be configured, uploaded, or transmitted to a router or a group of routers in any other possible way, as long as it can be stored in the classification rules tables of the forwarding engines of routers along the path of the flow to the final destination. If the flow label is used within a Differentiated Services framework, the values of the flow labels are preset or agreed upon, and specified in a Service Level Agreement (SLA), Service Level Specification (SLS), Traffic Conditioning Agreement (TCA), or Traffic Conditioning Specification (TCS) [Diffserv]. This model is typical of Differentiated Services.

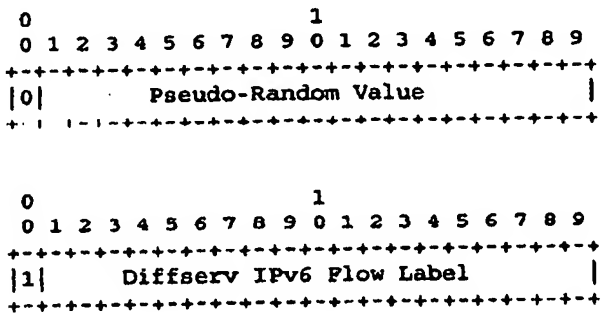
(e) In general, all packets belonging to the same flow are sent with the same source address, destination address, and flow label. However, flows can be trunked, or aggregated in macro-flows. The flows, members of a macro-flow, may have different source or destination addresses. The trunking, or aggregation of flows is achieved by simply wildcarding some bits or all bits in some of the fields of the multi-field classification rules, which contain source address, destination address, and flow label. In other words range addresses and/or flow labels can be used.

(f) The routers or destinations are permitted, but not required, to verify that these conditions are satisfied. If a violation is detected, it should be reported to the source by an ICMP Parameter Problem message, Code 0, pointing to the high-order octet of the Flow Label field (i.e., offset 1 within the IPv6 packet).

(g) The Diffserv flow labels to not have a time to live rule. However, changes to the value of a flow label of a flow, and/or the correspondent flow label classifier values MUST be synchronized. When the flow label value of a flow is changed, the change must be reflected in the change of the value of the flow label in the multi-Field flow label classifier.

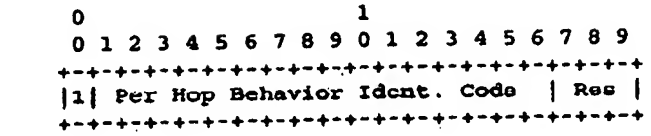
7.1 IPv6 Flow Label Format

In order to preserve compatibility with the random number method of selecting a flow label value defined in [IPv6], but relax that definition to allow a flow label format that would work with Diffserv, the following new format of the flow label could be used:



7.1.1 Diffserv IPv6 Flow Label Format

The Diffserv IPv6 Flow Label is a number that is constructed based on the Differentiated Services "Per Hop Behavior Identification Code" (PHB ID) [PHB ID]:



The "Res" bits are reserved.

Conforming to [PHB ID], the PHB ID is either directly derived from a standard differentiated services code point [DSCP-Def], or it is an "IANA Assigned Value". In either case, it captures the differentiated services treatment intended to be applied to the packet. Unlike the value of the traffic class field, it is not locally mapped and is therefore suitable for use in an end to end header field. Although it captures less specific information than the port numbers and protocol number normally used in an MF classifier, it nevertheless allows for MF classification at a differentiated service domain ingress.

7.1.2 Other Possible IPv6 Flow Label Formats

There are various other ways in which a Flow Label can be encoded, each way with its advantages and disadvantages. Several ideas of flow label encoding are enumerated in Appendix A.

7.2 Conceptual Model for Diffserv use of IPv6 Flow Label

Diffserv can be used in IPv6 access networks for IPv6 QoS of individual flows of traffic between users and the access networks. The nature of the contractual agreements between the users and the access network providers create an environment in which Diffserv with Multi-Field (M-F) classifiers could be easier to use, more efficient, and more practical as an alternative to Intserv and RSVP.

The IPv6 flow label classifier is basically a 3 element tuple - source and destination IPv6 addresses, and the IPv6 flow label [Diffserv-Flow-Label]. It is an alternative to the 5 element tuple (addresses, ports, and protocol). It helps the IPv6 flow label to achieve, as it is supposed, a more efficient processing of packets in quality of service engines in IPv6 forwarding devices.

Whether using algorithmic mapping of port numbers and protocol, IANA values, or just a number randomly chosen, the key for the flow label to work with Diffserv is that the "flow_label value" or range of values MUST be known, and agreed by two sides: the network client and the network provider. The "flow label value" is captured in SLAs, SLSs, TCAs, TCSs. For the mechanism to work several things have to

happen:

(1.) Packets leaving the client networks carry the correct flow label value. This can be achieved in several ways:

a. end-node IPv6 protocol stacks, and/or IPv6 applications can be configured with the flow label "value". The flow label "value" is set first by an application. If the application has not set a flow label "value", then the "value" is set by the protocol stack. The default values would be hard-coded in applications and protocol stacks, or could result from "algorithmic mapping", if such mappings exists. The default value could be zero, in which case the flow label would have no significance. According to this model, when packets are transmitted, end-nodes will force the correct flow label in the IPv6 headers of outgoing packets.

if a. is not TRUE, then

b. the first hop routers would have to force the correct flow label on packets leaving the network. To accomplish this role, these routers would be configured with MF classifiers. These routers would classify the traffic that is forwarded downstream from, and away from the originating end-nodes. The action subsequent to the classification would be to set the correct flow label in each packet. Classification on such a router's input line card, or interface would result, for the matching packets, in a correct flow label being forced in the IPv6 headers of packets when they are transmitted on the output interface or line card.

while it is likely that "b." would not be needed, "a." or "b." would provide the correct flow label in packets leaving the client's network.

(2.) Packets coming into the provider network can be policed based on flow label. The provider, based on the SLAs, SLSs, TCAs, TCSs agreed with the client, configures MF classifiers that look like:

C = (SA, SAPrefix, DA, DAPrefix, Flow-Label)

or

C' = (SA, SAPrefix, DA, DAPrefix, Flow-label-Min:FLow-label-Max)

Another representation of the classifier for example is:

```
Flow-label-classifier:
Type:                       IPv6-3-tuple
IPv6DestAddrValue:          1:2:3:4:5:6:7:8::1
IPv6DestPrefixLength:       128
IPv6SrcAddrValue:           8:7:6:5:4:3:2:1::2
IPv6SrcPrefixLength:        128
IPv6FlowLabel:              57
```

or

```
Flow-label-classifier:
Type:                       IPv6-3-tuple
IPv6DestAddrValue:          1:2:3:4:5:6:7:8::1
IPv6DestPrefixLength:       128
IPv6SrcAddrValue:           8:7:6:5:4:3:2:1::2
IPv6SrcPrefixLength:        128
IPv6FlowLabelMin:           1
IPv6FlowLabelMax:           57
```

and

```
Flow-label-classifier:
Type:                       IPv6-4-tuple
IPv6DestAddrValue:          1:2:3:4:5:6:7:8::1
IPv6DestPrefixLength:       128
IPv6SrcAddrValue:           8:7:6:5:4:3:2:1::2
IPv6SrcPrefixLength:        128
IPv6FlowLabel:              57
IPv6DSCP:                   28
```

or

```
Flow-label-classifier:
Type:                       IPv6-4-tuple
IPv6DestAddrValue:          1:2:3:4:5:6:7:8::1
IPv6DestPrefixLength:       128
IPv6SrcAddrValue:           8:7:6:5:4:3:2:1::2
IPv6SrcPrefixLength:        128
IPv6FlowLabelMin:           1
IPv6FlowLabelMax:           57
IPv6DSCP:                   28
```

The classifiers are configured in the network provider's edge routers, etc...

The classification engines in those routers would match packet header information to classification rules as follows:

Incoming packet header (SA, DA, Flow Label)
Match
Classification rules table entry (C or C')

From this step, the Diffserv processing continues the same way as for any other MF Classifier [Diffserv-Model].

8. Security Considerations

This document introduces no new security concerns when the pseudo-random flow label format is used. In the case of a diffserv flow label, the security concerns are essentially identical to those concerning the diffserv field (traffic class) itself, as outlined in [DSCP-Def], [Diffserv], and [Differv-Tun].

When IPv6 packets are encrypted using ESP Transport or Tunnel Mode [IPSec-ESP], the port and protocol numbers are hidden, but the flow label is not. Thus MF classification remains possible even for encrypted traffic.

9. IANA Considerations

The IPv6 flow label format specified in this document, is based on the Differentiated Services Per Hop Behavior Identification Code (PHB ID), specified in [PHB ID]. The PHB ID can be a IANA assigned number. [PHD ID] contains a "IANA Considerations Section", following guidelines stated in [CONS]. No additional IANA considerations have to be made.

10. Acknowledgments

Some of the ideas in this draft were discussed with Thomas Eklund, and Walter Weiss. Jochen Metzler reviewed the specification and provided good feedback. The continued scrutiny of Steve Deering helped refining the document.

11. References

[IPv6] S. Deering, R. Hinden, "Internet Protocol Version 6 Specification", RFC 2460, December 1998.

[Intserv] R. Braden, D. Clark, S. Shenker, "Integrated Services in

the Internet Architecture: an Overview", RFC 1633, June 1994.

[Diffserv] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Service", RFC 2475, December 1998.

[DSCP-Def] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[PHB-ID] D. Black, S. Brim, B. Carpenter, F. Le Faucheur, "Per Hop Behavior Identification Codes", RFC 3140, June 2001.

[Diffserv-Tun] D. Black, "Differentiated Services and Tunnels", RFC 2983, October 2000.

[Diffserv-PIB] M. Fine, K. McCloghrie, J. Seligson, K. Chan, S. Hahn, A. Smith, "Differentiated Services Policy Information Base", Work in Progress.

[DiffServ-MIB] F. Baker, K. Chan, A. Smith "Management Information Base for the Differentiated Services Architecture", Work in Progress.

[Diffserv-Model] Y. Bernet, S. Blake, A. Smith, D. Grossman, "An Informal Management Model for Diffserv Routers", Work in Progress.

[Diffserv-Flow-Label] A. Conta, B. Carpenter, "A Definition of a IPv6 Flow Label Classifier", Work in Progress.

[RSVP] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin. "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[MPLS-Arch] Rosen, E., Viswanathan, A., and Callon, R., "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[MPLS-LDP] L. Anderson, P. Doolan, N. Feldman, A. Fredette, R. Thomas, "Label Distribution Protocol", RFC 3036, January 2001.

[RSVP-TE] D. O. Awduche, L. Berger, D. Gan. Tony Li, Vijay Srinivasan, George Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", Work in progress.

[IPSec-ESP] S. Kent, R. Atkinson, "IP Encapsulating Security Protocol (ESP)", RFC 2406, November 1998.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[CONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, October 1998.

[Assign] Postel, J., etc.. "Assigned Numbers", STD 2, RFC 1700, October 1994.

12. Authors' Addresses

Alex Conta
Transwitch Corporation
3 Enterprise Drive
Shelton, CT 06484
USA
Email: aconta@txc.com

Brian Carpenter
IBM
c/o iCAIR
Suite 150
1890 Maple Avenue
Evanston, IL 60201
USA
Email: brian@hursley.ibm.com

Appendix A: Other Possible IPv6 Flow Label Formats

This section enumerates several ideas, each with its positive and negative aspects.

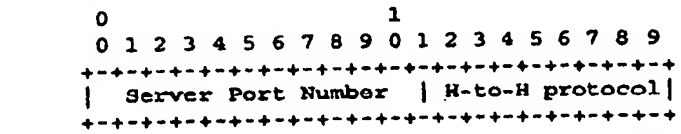
A possible solution to the issues discussed in section 5.7 is to compress or encode the host-to-host header information, and the host-to-host protocol type in the flow label value. This is an algorithmic mapping of the port numbers and protocol into the flow label. There are several ways in which this could be achieved, but only two are suggested in this section.

Another format mentioned further down in this section is one in which the length of the IPv6 headers helps locating in one step the host-to-host header for accessing the port information.

A.1 Server Port Format - Short Format

A possible solution to the issues discussed in section 5.7 is to compress or encode the host-to-host header information, and the host-to-host protocol type in the flow label value. This is an algorithmic mapping of the port numbers and protocol into the flow label. There are several ways in which this could be achieved, but only three are suggested in this section:

The Server Port Format is a format which is based on carrying in the flow label the server port number of a client/server application/communication.



The "Server Port Number" is the port number assigned to the server side of the client/server application. This provides an identification of the application, and the type of application, which is a quite good indication of the type of QoS characteristics needed for the traffic generated or accepted by that application. Obviously it does not provide the finer granularity within the use of one application on the same end-nodes, that the use of both source and destination ports provide. That is, it cannot differentiate among multiple instances of the same application running on the same two communicating end-nodes. But for Differentiated services purposes, it does not seem to really matter, since it is expected that the several instances of an application running on the same two end-nodes, would

generate or accept traffic which is of same category, class, or behavior.

The reduced number of bits (12 bits out of 16) limits the value to "IANA Well-known ports", that is ports from 1 to 1023, and a subset of "IANA registered ports" that is, from 1024 to 4095. Registered ports have values between 1024 and 65535 [Assign].

The "H-to-H protocol" is the host-to-host protocol identifier [Assign], that is, TCP, UDP, etc....

Advantage

The advantage of this flow label format is that the classification rule is the typical 5 or 6 tuple format of a Diffserv M-F Classifier [Diffserv-Model], containing the source, and destination addresses, the source and destination ports (in which one of the two is wildcard), the host to host protocol, and the DSCP field. So no new classification rule format is needed, and further, it is possible to aggregate parts of the IPv4, and IPv6 classification rules. Note that for classifying traffic in both directions, two classification rules must be configured. For instance a classification rule for TCP flows on port 80, between node A, and node B:

Source Address:A
Destination Address:B
Source Port:*
Destination Port:80
Host-to-Host Protocol 6 (TCP)

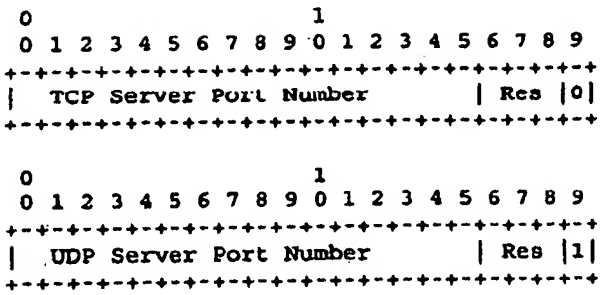
would be used for all traffic outgoing, from any port, to port 80.

Source Address:A
Destination Address:B
Source Port:80
Destination Port:*
Host-to-Host Protocol 6 (TCP)

would be used for all traffic outgoing from port 80, to any port.

A.2 Server Port Format - Long Format

Observation: Since TCP, and UDP are the two major host-to-host protocols that carry port numbers in their protocol headers, the field occupied by the "host-to-host" protocol could be reduced to 1 bit, indicating TCP or UDP, as it follows:
.sp

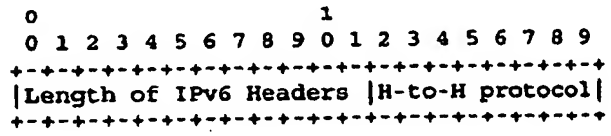


The "Res" bits are reserved.

The "TCP Server Port Number" or "UDP Server Port Number is the 16 bit port number assigned to the server side of the client/server application.

A.3 Header Length Format

Another possible solution to the issues discussed in section 5.7 is to store the IPv6 headers length, that is the length of the IPv6 main headers and IPv6 extensions headers preceding the host-to-host, or transport header. The length of the IPv6 headers in the flow label value would provide the information which a Diffserv QOS engine classifier could use to locate and fetch the source and destination ports, and apply those, along with the source and destination address and the host-to-host protocol from the flow label, to match the source and destination address, the source and destination ports and the protocol identifier elements of a Diffserv M-F classifier [Diffserv-Model].



The "Length of the IPv6 Headers" allows also skipping the IPv6 headers to access directly the host-by-host header for other purposes.

Additionally, this format is useful for classifying packets that are not TCP or UDP, and have no source and destination ports.

With this 12 bit encoding the maximum length of the IPv6 headers that

could be represented is 4Kbytes. However, the restriction on headers length can be significantly reduced. IPv6 headers are 8byte aligned, therefore the length could be represented as the number of 8byte chunks occupied by the headers, in which case the maximum length would be 32Kbytes.

If all of the above formats would be used, then there are two ways to separate this last type of encoding from the other two mentioned above:

(i) always use a signaling mechanism to distribute the flow label values, and so the type of the format would be stored as part of the flow state.

(ii) use a bit field to discriminate the formats. For instance, a two bit field could be used to indicate the first, second, or third type of format.

Note:

The suggestions described in this section are in fact an exploration of possible solutions. Each suggestion has advantages and disadvantages. They are kept in this section at least for recording purposes.